S&ZFH030508

Method and Arrangement for Arithmetic Encoding and Decoding Binary States and a Corresponding Computer Program and a Corresponding Computer-readable Storage Medium

Cross-Reference to Related Application:

This application is a continuation of copending International Application No. PCT/EP03/04654, filed May 2, 2003, which designated the United States and was not published in English.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to a method and an arrangement for arithmetically encoding and decoding binary states and to a corresponding computer program and a corresponding computer-readable storage medium which may in particular be used in digital data compression.

2. Description of the Related Art:

The present invention describes a new efficient method for binary arithmetic coding. There is a demand for binary arithmetic coding in most different application areas of digital data compression; here, in particular applications in the fields of digital image compression are of special interest. In numerous standards for image coding, like e.g. JPEG, JPEG-2000, JPEG-LS and JPIG, methods for a binary arithmetic coding were defined. Newer standardization activities make also the future use of such coding technologies obvious in the field of video coding (CABAC in H. 264/AVC) [1].

The advantages of arithmetic coding (AC) in contrast to the Huffman coding [2] which has up to now been used in practice, may basically be characterized by three features:

1. By using the arithmetic coding, by simple adaptation mechanisms a dynamic adaptation to the present source statistic may be obtained (adaptivity).

2. Arithmetic coding allows the allocation of a non-integer number of bits per symbol to be coded and is therefore suitable to achieve coding results which illustrate an approximation of the entropy as the theoretically given lower bound (entropy approximation) [3].

3. Using suitable context models statistical bindings between symbols for a further data reduction may be used with arithmetic coding (intersymbol redundancy) [4].

As a disadvantage of an application of the arithmetic coding, generally the increased calculation effort compared to Huffman coding is regarded.

The concept of the arithmetic coding goes back to the basic documentation for information theory by Shannon [5]. First conceptional construction methods were firstly published by Elias [6]. A first LIFO (last-in-first-out) variant of the arithmetic coding was designed by Rissanen [7] and later modified [8] [9] [10] by different authors to the FIFO implementations (first-in-first-out).

All of those documents have the basic principle of recursive partial interval decomposition in common. Corresponding to the given probabilities P("0") and P("1") of two results {"0", "1"} of a binary alphabet a primarily given interval, e.g. the interval [0, 1), is recursively decomposed into partial intervals depending on the

occurrence of individual events. Here, the size of the resulting partial interval as the product of the individual probabilities of the occurring events is proportional to the probability of the sequence of individual events. As every event S_i adds a contribution of $H(S_i) = -\log(P(S_i))$ of the theoretical information content $H(S_i)$ of S_i to the overall rate by the probability $P(S_i)$, a relation between the number N_{Bit} of bits for illustrating the partial interval and the entropy of the sequence of individual events results, which is given by the right side of the following equation:

$$N_{Bit} = -\log \prod_i P(Si) = -\sum_i \log P(Si)$$

The basic principle, however, first of all requires a (theoretically) unlimited accuracy in the illustration of the resulting partial interval and apart from that it has the disadvantage that only after the coding of the last result may the bits for a representation of the resulting partial interval be output. For practical application purposes it was therefore decisive to develop mechanisms for an incremental output of bits with a simultaneous representation with numbers of a predetermined fixed accuracy. These were first introduced in the documents [3] [7] [11].

In Fig. 1, the basic operations for a binary arithmetic coding are indicated. In the illustrated implementation the current partial interval is represented by the two values L and R, wherein L indicates the offset point and R the size (width) of the partial interval, wherein both quantities are respectively illustrated using b-bit integers. The coding of a bit $\in \{0, 1\}$ is thereby basically performed in five substeps: In the first step using the probability estimation the value of the less probable symbol is determined. For this symbol, also referred to as LPS (least probable symbol), in contrast to the MPS (most probable symbol), the probability estimation P_{LPS} is used in the

second step for calculating the width R_{LPS} of the corresponding partial interval. Depending on the value of the bit to be coded L and R are updated in the third step. In the forth step the probability estimation is updated depending on the value of the just coded bit and finally the code interval R is subjected to a so-called renormalization in the last step, i.e. R is for example rescaled so that the condition $R \in [2^{b-2}, 2^{b-1}]$ is fulfilled. Here, one bit is output with every scaling operation. For further details please refer to [10].

The main disadvantage of an implementation, as outlined above, now lies in the fact that the calculation of the interval width R_{LPS} requires a multiplication for every symbol to be coded. Generally, multiplication operations, in particular when they are realized in hardware, are cost- and time-intensive. In several research documents methods were examined to replace this multiplication operation by a suitable approximation [11] [12] [13] [14]. Hereby, the methods published with reference to this topic may generally be separated into three categories.

The first group of proposals for a multiplication-free, binary arithmetic coding is based on the approach to approximate the estimated probabilities P_{LPS} so that the multiplication in the second step of Fig. 1 may be replaced by one (or several) shift and addition operation(s) [11] [14]. For this, in the simplest case the probabilities P_{LPS} are approximated by values in the form of 2^{-q} with the integer $q > 0$.

In the second category of approximative methods it is proposed to approximate the value range of R by discrete values in the form $(1/2 - r)$, wherein $r \in \{0\} \cup \{2^{-k} \mid k > 0, k \text{ integer}\}$ is selected [15] [16].

The third category of methods is only known from the fact that here any arithmetic operations are replaced by table

accesses. To this group of methods on the one hand the Q-coder used in the JPEG standard and related methods, such as the QM- and MQ-coder [12], and on the other hand the quasi-arithmetic coder [13] belong. While the latter method performs a drastic limitation of the number b of bits used for the representation of R in order to obtain acceptably dimensioned tables, in the Q-coder the renormalization of R is implemented so that R may at least approximately be approximated by 1. This way the multiplication for determining R_{LPS} is prevented. Additionally, the probability estimation using a table in the form of a finite state machine is operated. For further details please see [12].

SUMMARY OF THE INVENTION

It is the object of the present invention to provide a method and an arrangement for an arithmetic encoding and decoding of binary states and a corresponding computer program and a corresponding computer-readable storage medium which eliminate the mentioned disadvantages and in particular (a) do not require a multiplication, (b) allow a probability estimation without calculation effort and (c) simultaneously guarantee a maximum coding efficiency over a wide range of typically occurring symbol probabilities.

In accordance with a first aspect, the present invention provides a method for an arithmetic encoding and decoding of binary states, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K $(1 \leq k \leq K)$ to every interval width R and one P_n $(1 \leq n \leq N)$ to every probability, and that in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding

process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

In accordance with a second aspect, the present invention provides an arrangement having at least one processor and/or chip, which is/are implemented such that a method for an arithmetic encoding and decoding of binary states is may be performed, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K ($1 \leq k \leq K$) to every interval width R and one P_n ($1 \leq n \leq N$) to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

In accordance with a third aspect, the present invention provides a computer program which enables a computer after it has been loaded into the storage of the computer to

perform a method for an arithmetic encoding and decoding of binary states, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K $(1 \leq k \leq K)$ to every interval width R and one P_n $(1 \leq n \leq N)$ to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K $(1 \leq k \leq K)$ and a representative probability state P_n $(1 \leq n \leq N)$ by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

In accordance with a fourth aspect, the present invention provides A computer-readable storage medium on which a computer program is stored which enables a computer after it has been loaded into the storage of the computer to perform a method for an arithmetic encoding and decoding of binary states, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K $(1 \leq k \leq K)$ to every interval width R and one P_n $(1 \leq n \leq N)$ to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding

process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

One method for an arithmetic encoding and decoding of binary states is advantageously performed so that in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K ($1 \leq k \leq K$) to every interval width R and one P_n ($1 \leq n \leq N$) to every probability, and that in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations.

Another preferred implementation of the invention is characterized by the fact that based on the interval currently to be evaluated with a width R for determining the associated interval width Q_K an index q_index is

determined by a shift and bit masking operation applied to the computer-internal/binary representation of R.

It is also advantageous when based on the interval currently to be evaluated with a width R for determining the associated interval width Q_K an index q_index is determined by a shift operation applied to the computer-internal/binary representation of R and a downstream access to a table Qtab, wherein the table Qtab contains the indices of interval widths which correspond to the values of R which were pre-quantized by the shift operation.

It is in particular advantageous when the probability estimation underlying the symbol to be encoded or decoded is associated to a probability state P_n using an index p_state.

It is also an advantage when the determination of the interval width R_{LPS} corresponding to the LPS is performed by an access to the table Rtab, wherein the table Rtab contains the values corresponding to all K quantized values of R and to the N different probability states of the interval width R_{LPS} as product values ($Q_K * P_n$). The calculation effort is reduced in particular when the determination of the interval width R_{LPS} corresponding to the LPS is performed by an access to the table Rtab, wherein for evaluating the table the quantization index q_index and the index of the probability state p_state are used.

It is further provided that in the inventive method for the N different representative probability states transition rules are given, wherein the transition rules indicate which new state is used based on the currently encoded or decoded symbol for the next symbol to be encoded or decoded. It is hereby of an advantage when a table Next_State_LPS is created which contains the index m of the new probability state P_m when a least probable symbol (LPS)

occurs in addition to the index n of the currently given probability state P_n, and/or when a table Next_State_MPS is created which contains the index m of the new probability state P_m when a most probable symbol (MPS) occurs in addition to the index n of the currently given probability state P_n.

An optimization of the method for a table-aided binary arithmetic encoding and decoding is achieved in particular by the fact that the values of the interval width R_{LPS} corresponding to all K interval widths and to all N different probability states are filed as product values ($Q_K * P_n$) in a table Rtab.

A further optimization is achieved when the number K of the quantization values and/or the number N of the representative states are selected depending on the preset accuracy of the coding and/or depending on the available storage room.

One special implementation of the encoding in the inventive method includes the following steps:

1. Determination of the LPS
2. Quantization of R:
   q_index = Qtab[R>>q]
3. Determination of R_{LPS} and R:
   R_{LPS} = Rtab [q_index, p_state]
   R = R − R_{LPS}
4. Calculation of the new partial interval:
   if (bit : LPS) **then**
   L ← L + R
   R ← R_{LPS}
   p_state ← Next_State_LPS [p_state]
   if (p_state = 0) then valMPS ← 1 − valMPS
   else
   p_state ← Next_State_MPS [p_state]

5. Renormalization of L and R, writing bits, wherein

q_index	describes the index of a quantization value read out of Qtab,
p_state	describes the current state,
$\mathbf{R_{LPS}}$	describes the interval width corresponding to the LPS and
valMPS	describes the bit corresponding to the MPS.

The decoding in a special implementation of the inventive method includes the following steps:

1. Determination of the LPS
2. Quantization of R:

   q_index = Qtab[R>>q]
3. Determination of R_{LPS} and R:

   $\mathbf{R_{LPS}}$ **= Rtab [q_index, p_state]**

   R = R – $\mathbf{R_{LPS}}$
4. Determination of bit depending on the position of the partial interval:

```
if (V ≥ R) then
  bit ← LPS
  V ← V - R
  R ← R_LPS
      if (p_state = 0) valMPS ← 1 - valMPS
      p_state ← Next_State_LPS [p_state]
else
      bit ← MPS
      p_state ← Next_State_MPS [p_state]
```

5. Renormalization of R, reading out one bit and updating V, wherein

q_index	describes the index of a quantization value read out of Qtab,
p_state	describes the current state,
$\mathbf{R_{LPS}}$	describes the interval width corresponding to the LPS,
valMPS	describes the bit corresponding to the MPS, and
V	describes a value from the interior of the

current partial interval.

In another special implementation of the inventive method it is provided that in encoding and/or decoding the calculation of the quantization index q_index is performed in the second substep after the calculation regulation:

q_**index** = (R >> q) & Qmask

wherein Qmask illustrates a bit mask suitably selected depending on K.

If a uniform probability distribution is present a further optimization of the method for a table-aided binary arithmetic encoding and decoding may be achieved by the fact that in the encoding according to claim 12 the substeps 1 to 4 are performed according to the following calculation regulation:

R ← R >> 1
if (bit = 1) **then**
    L ← L + R

or

that the substeps 1 to 4 of the encoding according to claim 12 are performed according to the following calculation regulation:

L ← L << 1
if (bit = 1) **then**
    L ← L + R

and wherein in the last alternative the renormalization (substep 5 according to claim 12) is performed with doubled decision threshold values and no doubling of L and R is performed, and

that in the decoding according to claim 13 the substeps 1 to 4 are performed according to the following calculation regulation:

R ← R >>1
if (V ≥ R) **then**
    bit ← 1

```
        V ← V - R
    else
        bit ← 0,
```

or

the substeps 1 to 5 of the decoding according to claim 13 are performed according to the following calculation regulation:

1. Reading out one bit and updating V
2. Determination of bit according to the position of the partial interval:

```
   if (V ≥ R) then
           bit ← 1
           V ← V - R
   else
           bit ← 0.
```

It further turns out to be advantageous when the initialization of the probability models is performed depending on a quantization parameter SliceQP and preset model parameters m and n, wherein SliceQP describes the quantization parameter preset at the beginning of a slice and m and n describe the model parameters.

It is also advantageous when the initialization of the probability models includes the following steps:

```
1. preState = min(max(1, ((m * SliceQP) >>4)+n), 2*N)
2. if (preState <=N) then
           p_state = N+1 - preState
           valMPS = 0
   else
           p_state = preState - N
           valMPS = 1,
```

wherein valMPS describes the bit corresponding to the MPS, SliceQP describes the quantization parameter preset at the beginning of a slice and m and n describe the model parameters.

One arrangement for an arithmetic encoding and decoding of binary states includes at least one processor which is/are implemented such that a method for an arithmetic encoding and decoding may be performed, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K $(1 \leq k \leq K)$ to every interval width R and one P_n $(1 \leq n \leq N)$ to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K $(1 \leq k \leq K)$ and a representative probability state P_n $(1 \leq n \leq N)$ by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying the symbol to be encoded or to be decoded according to the given allocation regulations..

One computer program for an arithmetic encoding and decoding of binary states allows a computer, after it has been loaded into the storage of the computer, to perform an method for an arithmetic encoding and decoding, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K $(1 \leq k \leq K)$ to every interval width R and one P_n $(1 \leq n \leq N)$ to every probability, and wherein in a second step the encoding or decoding of the binary states take place by

performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively, using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying for the symbol to be encoded or to be decoded according to the given allocation regulations.

For example, such computer programs may be provided (against a certain fee or for free, freely accessible or password-protected) which may be downloaded into a data or communication network. The thus provided computer programs may then be made useable by a method in which a computer program according to claim 22 is downloaded from a network for data transmission, like for example from the internet to a data processing means connected to the network.

For performing a method for an arithmetic encoding and decoding of binary states preferably a computer-readable storage medium is used on which a program is stored which allows a computer, after it has been loaded into the storage of the computer, to perform a method for an arithmetic encoding or decoding, wherein in a first step a presetable value range for the specification of the interval width R is separated in K representative interval widths $\{Q_1, \ldots, Q_K\}$, a presetable value range for the specification of the probabilities is separated in N representative probability states $\{P_1, \ldots, P_N\}$ and allocation regulations are given, which allocate one Q_K ($1 \leq k \leq K$) to every interval width R and one P_n ($1 \leq n \leq N$) to every probability, and wherein in a second step the encoding or decoding of the binary states take place by performing the calculation of the new interval width to be derived in the encoding or decoding process, respectively,

using a representative interval width Q_K ($1 \leq k \leq K$) and a representative probability state P_n ($1 \leq n \leq N$) by arithmetic operations other than multiplication and division, wherein the representative interval width Q_K is determined by the basic basis interval of the width R and the representative probability state P_n is determined by the probability estimation underlying as the basis for the symbol to be encoded or to be decoded according to the given allocation regulations.

The new method is distinguished by the combination of three features. First of all, similar to the Q-coder the probability estimation is performed using a finite state machine (FSM), wherein the generation of the N representative states of the FSM is performed offline. The corresponding transition rules are thereby filed in the form of tables.

A second characteristic feature of the invention is a prequantization of the interval width R to a number of K predefined quantization values. This allows, with a suitable dimensioning of K an N, the generation of a table which contains all K x N combinations of precalculated product values R x P_{LPS} for a multiplication-free determination of R_{LPS}.

For the use of the presented invention in an environment in which different context models are used among which also such with (almost) uniform probability distribution are located, as an additional (optional) element a separated branch is provided within the coding machine in which assuming an equal distribution the determination of the variables L and R and the renormalization regarding the calculation effort is again substantially reduced.

As a whole the invention in particular provides the advantage that it allows a good compromise between a high

coding efficiency on the one hand and a low calculating effort on the other hand.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and features of the present invention will become clear from the following description taken in conjunction with the accompanying drawings, in which:

Fig. 1	shows an illustration of the basic operations for a binary arithmetic coding;
Fig. 2	shows a modified scheme for a table-aided arithmetic encoding;
Fig. 3	shows the principle of the table-aided arithmetic decoding;
Fig. 4	shows the principle of encoding or decoding, respectively, binary data having a uniform distribution;
Fig. 5	shows an alternative realization of encoding or decoding, respectively, for binary data with a uniform distribution; and
Fig. 6	shows the initialization of the probability models depending on a quantization parameter SliceQP and preset model parameters m and n.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

First of all, however, the theoretical background is to be explained in more detail:

Table-aided probability estimation

As it was already mentioned above, the effect of the arithmetic coding relies on an estimation of the occurrence probability of the symbols to be coded which is to be as good as possible. In order to enable an adaptation to non-

stationary source statistics, this estimation needs to be updated in the course of the coding process. Generally, usually methods are used for this which operate using scaled frequency counters of the coded results [17]. If C_{LPS} and C_{MPS} designates counters for the occurrence frequencies of LPS and MPS, then using these counters the estimation

$$P_{LPS} = \frac{C_{LPS}}{C_{LPS} + C_{MPS}} \tag{1}$$

may be performed and then the operation outlined in Fig. 1 of the interval separation may be carried out. For practical purposes the division required in equation (1) is disadvantageous. It is often convenient and required, however, to perform a rescaling of the counter readings when a predetermined threshold value C_{max} of the overall counter $C_{Total} = C_{MPS} + C_{LPS}$ is exceeded. (In this context it is to be noted that with a b-bit representation of L and R the smallest probability which may be indicated correctly is 2^{-b+2}, so that for preventing that this lower limit is fallen short of, if necessary a rescaling of the counter readings is required.) With a suitable selection of C_{max} the reciprocal values of C_{Total} may be tabulated, so that the division required in equation (1) may be replaced by a table access and by a multiplication and shift operation. In order to prevent also these arithmetic operations, however, in the present invention a completely table-aided method is used for the probability estimation.

For this purpose in a training phase representative probability states $\{P_K \mid 0 \leq k < N_{max}\}$ are preselected, wherein the selection of the states is on the one hand dependent on the statistics of the data to be coded and on the other hand on the side conditions of the default maximum number N_{max} of states. Additionally, transition rules are defined which indicate which new state is to be used for the next symbol to be coded based on the currently coded symbol. These transition rules are provided in the

form of two tables: $\{Next\_State\_LPS_k \mid 0 \leq k < N_{max}\}$ and $\{Next\_State\_MPS_k \mid 0 \leq k < N_{max}\}$, wherein the tables provide the index m of the new probability state P_m when an LPS or MPS occurs, respectively, for the index n of the currently given probability state. It is to be noted here, that for a probability estimation in the arithmetic encoder or decoder, respectively, as it is proposed herein, no explicit tabulation of the probability states is required. Rather, the states are only implicitly addressed using their respective indices, as it is described in the following section. In addition to the transition rules it needs to be specified at which probability states the value of the LPS and MPS needs to be exchanged. Generally, there will only be one such excellent state which may be identified using its index p_state.

Table-aided interval separation

Fig. 2 shows the modified scheme for a table-aided arithmetic coding, as it is proposed herein. After the determination of the LPS, first of all the given interval width R is mapped to a quantized value Q using a tabulated mapping Qtab and a suitable shift operation (by q bit). Alternatively, the quantization may in special cases also be performed without the use of a tabulated mapping Qtab only with the help of a combination of shift and masking operations. Generally, here a relatively coarse quantization to K = 2 … 8 representative values is performed. Also here, similar to the case of the probability estimation, no explicit determination of Q is performed; rather, only an index q_index is transferred to Q. This index is now used together with the index p_state for a characterization of the current probability state for the determination of the interval width R_{LPS}. For this, now the corresponding entry of the table Rtab is used. There, the $K \cdot N_{max}$ product values $R \times P_{LPS}$, that correspond to all K quantized values of R and the N_{max} different from the probability states, are entered as integer values with an

accuracy of generally b-2 bits. For practical implementations a possibility is given here to weigh up between the storage requirements for the table size and the arithmetic accuracy which finally also determines the efficiency of the coding. Both target variables are determined by the granularity of the representation of R and P_{LPS}.

In the forth step of Fig. 2 it is shown, how the updating of the probability state p_state is performed depending on the above coded event bit. Here, the transition tables Next_State_LPS and Next_State_MPS are used which were already mentioned above in the section "table-aided probability estimation". These operations correspond to the updating process indicated in Fig. 1 in step 4 which is not explained in more detail.

Fig. 3 shows the corresponding flow chart of the table-aided arithmetic decoding. For characterizing the current partial interval in the decoder the interval width R and a value V is used. The latter is present within the partial interval and is refined successively with every read-out bit. As it may be seen from Fig. 3, the operations for the probability estimation and the determination of the interval width R are performed according to those of the encoder.

Coding with uniform probability distribution

In applications in which e.g. signed values are to be coded whose probability distribution is arranged symmetrically around zero, for coding the sign information generally an equal distribution may be assumed. As this information is one the one hand to be embedded in the arithmetic bit stream, while it is on the other hand not sensible to use a relatively compact apparatus of the table-aided probability estimation and interval separation for the case of a probability of $p \approx 0.5$, it is for this special case

proposed to optionally use a special encoder/decoder procedure which may be illustrated as follows.

In this special case the interval width of the new partial interval may be determined in the encoder by a simple shift operation corresponding to a bisection of the width of the original interval R. Depending on the value of the bit to be coded, the upper or lower half of R, respectively, is then selected as a new partial interval (see Fig. 4). The subsequent renormalization and output of bits is performed as in the above case of the table-aided solution.

In the corresponding decoder the required operations are reduced to determining the bit to be decoded using the value of V relatively to the current interval width R by a simple comparison operation. In the case that the decoded bit is set, V is to be reduced by the amount of R. As it is illustrated in Fig. 4, the decoding is ended by the renormalization and updating of V using the bit to be read in next.

An alternative realization of the coding of events with a uniform probability distribution is illustrated in Fig. 5. In this exemplary implementation the current interval width R is not modified. Instead, V is first doubled by a shift operation in the encoder. Depending on the value of the bit to be coded, then, similar to the above example, the upper or lower half, respectively, of R is selected as a new partial interval (see Fig. 5). The subsequent renormalization and output of bits is performed as in the above case of the table-aided solution with the difference that the doubling of R and L is not performed and that the corresponding comparison operations are performed with doubled threshold values.

In the corresponding decoder of the alternative realization first of all a bit is read out and V is updated. The second step is performed in the same way as step 1 in Fig. 4, i.e.

the bit to be decoded is determined using the value of V relative to the current interval width R by a simple comparison operation, and in the case in which the decoded bit is set, V is to be reduced by the amount of R (see Fig. 5).

Addressing and initializing the probability models

Every probability model, as it is used in the proposed invention, is indicated using two parameters: 1) The index p_state that characterizes the probability state of the LPS, and 2) the value valMPS of the MPS. Each of these two variables needs to be initialized at the beginning of the encoding or decoding, respectively, of a completed coding unit (in applications of video coding about one slice). The initialization values may thereby be derived from control information, like e.g. the quantization parameter (of a slice), as it is illustrated as an example in Fig. 6.

Forward-controlled initialization process

A further possibility of adaptation of the starting distributions of the models is provided by the following method. In order to guarantee a better adaptation of the initializations of the models, in the encoder a selection of predetermined starting values of the models may be provided. These models may be combined into groups of starting distributions and may be addressed using indices, so that in the encoder the adaptive selection of a group of starting values is performed and is transmitted to the decoder in the form of an index as page information. This method is referred to as a forward-controlled initialization process.

While this invention has been described in terms of several preferred embodiments, there are alterations, permutations, and equivalents which fall within the scope of this invention. It should also be noted that there are many

alternative ways of implementing the methods and compositions of the present invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

References

[1] T. Wiegand, G. Sullivan, *"Draft Text of Final Draft International Standard (FDIS) of Joint Video Specification (ITU-T Rec. H. 264 /ISO/IEC 14496-10 AVC)"*, JVT-G050, March 2003.

[2] D. A. Huffman, "A Method for Construction of Minimum Redundancy code", Proc. IRE, Vol. 40, pp. 1098-1101, 1952.

[3] I. H. Witten, R. M. Neal, J. G. Cleary, "Arithmetic Coding for Data Compression", Communication of the ACM, Vol. 30, No. 6, pp. 520-540, 1987.

[4] G. G. Langdon, J. Rissanen, "A Simple General Binary Source code", IEEE Transactions on Information Theory, Vol. 28, pp. 800-803, 1982.

[5] C. E. Shannon, "A Mathematical Theory of Communication", Bell Syst. Tech. Journal, vol. 27, pp. 379-423, 623-656, 1948.

[6] P. Elias, in "Information Theory and Coding", N. Abramson (Ed.), New York, Mc-Gra-Hill, 1963.

[7] J. Rissanen, "Generalized Kraft Inequality and Arithmetic Coding", IBM J. Res. Develop., Vol. 20, pp. 198-203, 1976.

[8] R. C. Pasco, "Source Coding and Algorithms for Fast Data Compression", Ph. D. Dissertation, Stanford University, USA, 1976.

[9] G. G. Langdon, "An Introduction to Arithmetic Coding", IBM J. Res. Develop., Vol. 28, pp. 135-149, 1984.

[10] A. Moffat, R. M. Neal, I. H. Witten, "Arithmetic Coding Revisited", Proc. IEEE Data Compression Conference, Snowbird (USA), pp. 202-211, 1996.

[11] J. Rissanen, K. M. Mohiuddin, "A Multiplication-Free Multialphabet Arithmetic Arithmetic Code", IEEE Trans. on Communication, Vol. 37, pp. 93-98, 1989.

[12] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, R. B. Arps, "An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder", IBM J. Res. Develop., Vol. 32, pp. 717-726, 1988.

[13] P. G. Howard, J. S. Vitter, "Practical Implementations of Arithmetic Coding", in "Image and Text Compression", J. Storer (Ed.), Norwell (USA), Kluwer, 1992.

[14] L. Huynh, A. Moffat, "A Probability-Ratio Approach to Approximate Binary Arithmetic Coding", IEEE Trans. on Information Theory, Vol. 43, pp. 1658-1662, 1997.

[15] D. Chevion, E. D. Karnin, E. Walach, "High-Efficiency, Multiplication Free Approximation of Arithmetic Coding", Proc. IEEE Data Compression Conference, Snowbird (USA), pp. 43-52, 1991.

[16] G. Feygin, P. G. Gulak, P. Chow, "Minimizing Excess Code Length and VLSI Complexity in the Multiplication Free Approximation of Arithmetic Coding", Inform. Proc. Manag., vol. 30, pp. 805-816, 1994.

[17] D. L. Duttweiler, Ch. Chamzas, "Probability Estimation in Arithmetic and Adaptive-Huffman Entropy Coders", IEEE Trans. on Image Processing, Vol. 4, pp. 237-246, 1995.